

НИУ ИТМО  
Факультет точной механики и технологий  
Кафедра технологии приборостроения

## Документация на программу «Нахождение квадратов и кубов чисел»

Выполнила  
студентка группы 1652  
Путинцева А.А.

Руководитель  
Афанасьев М.Я.

Санкт-Петербург, 2012 год

## Описание программы

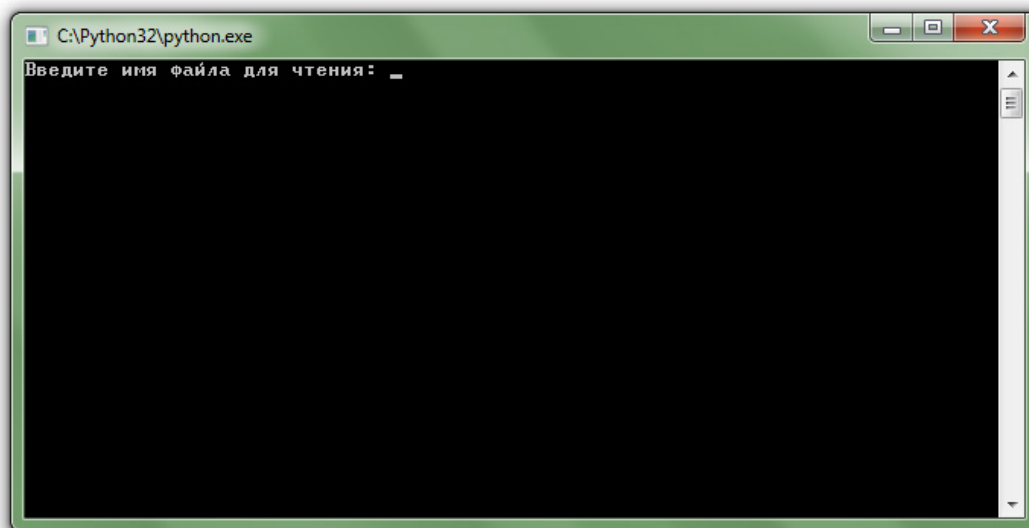
Представленная программа демонстрирует такие базовые навыки программирования на языке Python 3.2, как работа с числовыми данными, файлами (чтение из файла и запись в файл), обработка исключений и вывод полученных результатов на экран.

**Назначение программы:** вычисление квадратов и кубов чисел, записанных в файле с расширением .txt, вывод результатов на экран и их запись в файл.

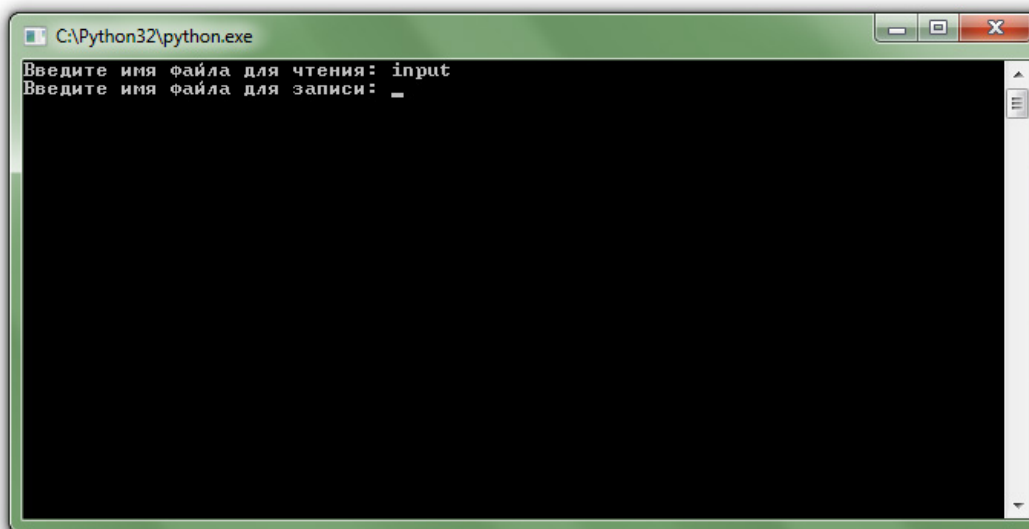
### *Инструкция для пользователя*

1. Запустите **square\_cube.py**.
2. В появившемся окне введите имя текстового файла с исходными данными (без расширения). Программой будет открыт файл <имя файла>.txt, расположенный в той же директории, что и программа. Убедитесь, что этот файл существует.

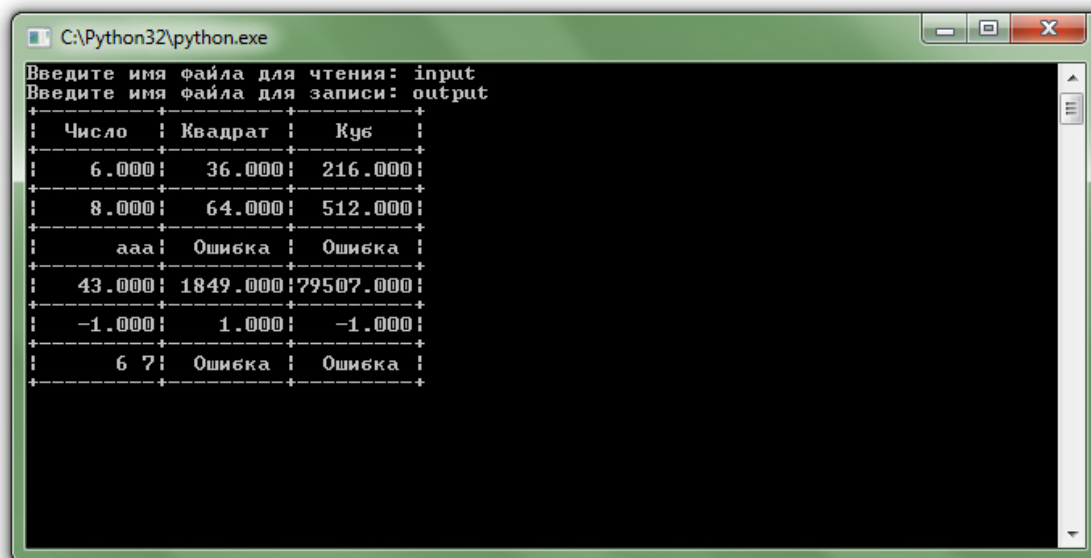
*Примечание.* Если файл с данными и исполняемый файл программы находятся в разных папках, необходимо прописать полный путь к файлу с данными. При этом расширение также не указывается.



3. Следующим шагом необходимо ввести имя файла, в который будет произведена запись.



4. Программа считывает все данные из файла и выводит полученные результаты на экран.



### *Пояснения к программе*

#### **Краткий алгоритм:**

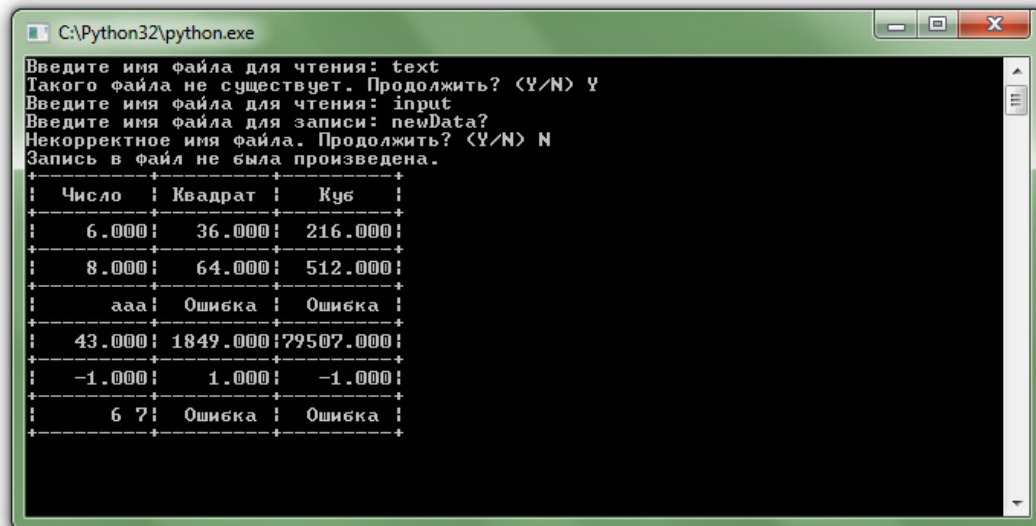
1. Запрос имени файла с исходными данными, проверка его корректности и открытие этого файла.
2. Чтение данных из открытого файла.
3. Создание файла для записи с именем, введенным пользователем (также реализуется осуществление проверки корректности введенного имени).
4. Формирование заголовка таблицы результатов, его вывод на экран и запись в файл.
5. Обработка исходных данных и формирование таблицы с параллельным выводом результатов на экран и записью в файл.
6. Закрытие всех открытых файлов.

#### **Обработка исключений:**

В программе реализована обработка таких исключений, как `IOError`, `ValueError` и `NameError`. Рассмотрим вышеупомянутые исключения подробнее:

- *ValueError* — поднимается, когда встроенная операция или функция получают параметр, который имеет правильный тип, но несоответствующее значение. В данном случае может возникнуть, если в исходном файле вместо числа в какой-либо строке (строках) введены некорректные данные (например, текст или несколько чисел в одной строке).
- *NameError* — возникает, когда локальное или глобальное имя не найдено. В рассматриваемой программе произойдет, если пользователь не введет корректное имя файла (и запись в файл не будет произведена), т.к. программа попытается закрыть несуществующий файл.

- *IOError* — исключение, связанное с ошибками ввода/вывода. В описываемой программе может произойти, если пользователь произведет попытку чтения данных из несуществующего файла либо задаст некорректное имя файла для записи результатов (например, содержащее недопустимые символы — \*, |, \, :, ", <, >, ?, /).



```
C:\Python32\python.exe
Введите имя файла для чтения: text
Такого файла не существует. Продолжить? (Y/N) Y
Введите имя файла для чтения: input
Введите имя файла для записи: newData?
Некорректное имя файла. Продолжить? (Y/N) N
Запись в файл не была произведена.

+-----+
| Число | Квадрат | Куб |
+-----+
| 6.000 | 36.000 | 216.000 |
+-----+
| 8.000 | 64.000 | 512.000 |
+-----+
|   ааа | Ошибка | Ошибка |
+-----+
| 43.000 | 1849.000 | 79507.000 |
+-----+
| -1.000 | 1.000 | -1.000 |
+-----+
| 6 7 | Ошибка | Ошибка |
+-----+
```

# Приложение

## Листинг программы

```
pathIn = False; # Флаг корректности введенного имени файла для чтения.

# Открытие файла с данными.
while (not pathIn):
    try:
        name = input("Введите имя файла для чтения: ");
        name += ".txt";
        file = open(name);
        pathIn = True;
    except IOError:
        pathIn = False;
        answ = input("Такого файла не существует. Продолжить? (Y/N) ");
        if (answ == 'N' or answ == 'n'):
            print("Работа программы прекращена пользователем.");
            break;
        if (answ != 'Y' and answ != 'y'):
            print("Введена неизвестная команда. Работа программы будет продолжена.");

if (pathIn):
    # Считывание данных из файла.
    arr = file.readlines();
    n = len(arr);

    # Открытие файла для записи.
    pathOut = False; # Флаг корректности введенного имени файла для записи.
    while (not pathOut):
        try:
            name = input("Введите имя файла для записи: ");
            name += ".txt";
            fileNew = open(name, "w");
            pathOut = True;
        except IOError:
            pathOut = False;
            answ = input("Некорректное имя файла. Продолжить? (Y/N) ");
            if (answ == 'N' or answ == 'n'):
                print("Запись в файл не была произведена.");
                break;
            if (answ != 'Y' and answ != 'y'):
                print("Введена неизвестная команда. Работа программы будет продолжена.");

    # Вывод заголовка таблицы на экран и его запись в файл.
    sep = "+" + 9 * "-" + "+" + 9 * "-" + "+" + 9 * "-" + "+";
    print(sep);
    print("| Число | Квадрат | Куб |");
    print(sep);
    if (pathOut): # Запись в файл.
        fileNew.write(sep + "\n");
        fileNew.write("| Число | Квадрат | Куб | \n");
        fileNew.write(sep + "\n");

# Обработка исходных данных и формирование таблицы.
for i in range (n):
    try:
        arr[i] = float(arr[i]);
        s = "{0:9.3f}|{1:9.3f}|{2:9.3f}|".format(arr[i], arr[i]**2,
arr[i]**3);
    except ValueError:
        s = "{0:>9}|".format(arr[i].replace('\n', '')) + " Ошибка |
```

```
Ошибка |";
    print(s);
    print(sep);
    if (pathOut): # Запись в файл.
        fileNew.write(s + "\n");
        fileNew.write(sep + "\n");

    # Закрытие файлов.
    try:
        file.close();
        fileNew.close();
    except NameError:
        pathIn = False;

input();
```